

High accuracy navigation in unknown environment using adaptive control

Fernando Ribeiro, Ivo Moutinho, Nino Pereira, Fernando Oliveira,
José Fernandes, Nuno Peixoto, Antero Salgado

Grupo de Automação e Robótica, Departamento de Electrónica Industrial,
Universidade do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
fernando@dei.uminho.pt, ivomauro@sarobotica.pt, ninopereira@sarobotica.pt,
fremaxeii@gmail.com, jose Luisfernandes@gmail.com, nunopeixoto35242@hotmail.com,
anterosalgado@aeiou.pt

Abstract. Aiming to reduce cycle time and improving the accuracy on tracking, a modified adaptive control was developed, which adapts autonomously to changing dynamic parameters. The platform used is based on a robot with a vision based sensory system. Goal and obstacles angles are calculated relatively to robot orientation from image processing software. Autonomous robots are programmed to navigate in unknown and unstructured environments where there are multiple obstacles which can readily change their position. This approach underlies in dynamic attractor and repulsive forces. This theory uses differential equations that produce vector fields to control speed and direction of the robot. This new strategy was compared with existing PID method experimentally and it proved to be more effective in terms of behaviour and time-response. Calibration parameters used in PID control are in this case unnecessary. The experiments were carried out in robot Middle Size League football players built for RoboCup. Target pursuit, namely, ball, goal or any absolute position, was tested. Results showed high tracking accuracy and rapid response to moving targets. This dynamic control system enables a good balance between fast movements and smooth behaviour.

1. Introduction

A dynamic approach was employed to control the movement of an autonomous robot which is meant to navigate towards a moving target or goal, avoiding obstacles and collisions. Navigation direction, ϕ , is a behaviour variable which varies from 0° to 360° relatively to an external reference.

In Fig. 1 is represented the robots' navigation direction ϕ_{robot} , as well as target direction ψ_{target} and obstacle direction ψ_{obs} . The target direction is the desired value to navigation direction. The direction of the obstacle is the erroneous direction, which must be avoided by the robot [1].

The robot movement is generated by continuously calculating values for the navigation direction. The time series $\phi(t)$ is generated by a dynamic system based on a differential equation in which the state variable is the robots' navigation direction:

$$\frac{d\varphi_{robot}}{dt} = f(\varphi_{robot}) = f_{target}(\varphi_{robot}) + f_{obs}(\varphi_{robot}) \quad (1)$$

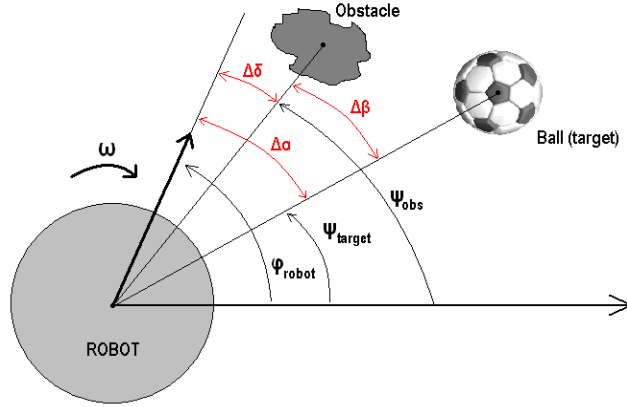


Fig. 1. Navigation direction of the robot depending on target and obstacle directions

The vector field of this dynamic system is build from a certain number of additive forces, each one specifying a particular value (attractive or repulsive) for the navigation direction. Each of these forces is characterized by a singular direction value, ψ_{target} or ψ_{obs} , intensity of attraction or repulsion and also by the range of direction values affected by them. A force by itself establishes an attractor, state asymptotically stable, or a repulsor, state asymptotically instable in the dynamics of navigation direction.

An attractive force, f_{target} , is used to pull the system to the desired value, namely to the target direction. On the other hand, a repulsive force, f_{obs} , ensures that the system avoids moving towards the obstacle direction. Summing all these forces, results in a non-linear dynamic system [2].

Since all angles are measured related to an external reference axis, the contributions of obstacles and target to the dynamic system do not depend on the actual orientation of the robot. This navigation control system was implemented and experimental essays were carried out on an autonomous robot.

2. Navigation Direction Control

2.1 Reaching a target

The differential equation that describes the behaviour of the system to pursuit a target is:

$$\frac{d\varphi_{robot}}{dt} = -k \cdot \tanh((\varphi_{robot} - \psi_{target}) \cdot \gamma) \quad (2)$$

where:

- φ_{robot} – Navigation direction of the robot
- ψ_{target} – Target direction.
- k – Maximum value of the attractive force
- γ – Target attraction intensity

The choice of this equation is not arbitrary, because the function of hyperbolic tangent has only one zero, the attractor (fixed point), and stabilizes at maximum and minimum values for the direction variation of the robot in time.

As the robot moves, the target direction varies, pulling the attractor over the possible values for the navigation direction, as represented in Fig. 2.

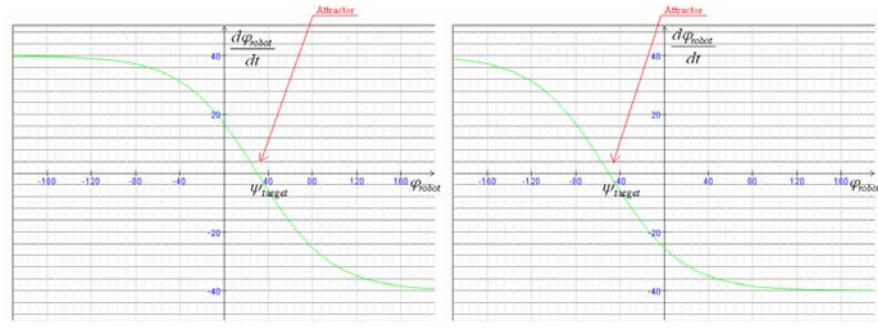


Fig. 2. Fixed point localization variation (attractor), as a function of the robot movement and target displacement

The rate of the direction variation in time corresponds to the angular velocity or robot rotational movement. The behaviour variable used for calculations was obtained from the difference between the robot direction angle and the target angle.

$$\Delta\alpha = \varphi_{robot} - \psi_{target} \quad (3)$$

Thus, the differential equation assumes the form:

$$\omega(t) = \frac{d\varphi_{robot}}{dt} = -\omega_{max} \cdot \tanh(\Delta\alpha \cdot \gamma) \quad (4)$$

where:

- ω – Angular velocity of the robot
- $\Delta\alpha$ – Target angle relative to the robot
- ω_{\max} – Maximum rotation velocity
- γ – Target Attraction Intensity

In this way, there isn't any shift of the attractor fixed point, since what really matters is the difference between the robot navigation direction and the target direction. This difference should always take the value zero.

No matter the navigation direction or the target direction, every time the difference assumes a non-zero value, the system acquires a positive or negative angular velocity which decreases as the difference becomes null. At that time the system stabilizes and the robot stops. This fixed point is stable and is denominated attractor since the direction variation rate is negative at that point and the system tends to the desired value. The angular velocity shown in the picture above (ω), is an entry parameter to the robot command functions that control the motors and varies from 0 to 40.

A very important parameter is γ , since it allows the specification of the intensity of the target attraction. In other words, it defines the speed variation of the angular velocity, or system time-response. A special attention must be given to the parameter γ since high values could make the system unstable.

2.2 Obstacle avoidance

The differential equation which describes the system behaviour for obstacle avoidance is given by:

$$\frac{d\varphi_{robot}}{dt} = -\lambda \cdot (\varphi_{robot} - \psi_{obs}) \cdot e^{-\frac{(\varphi_{robot} - \psi_{obs})^2}{2\delta^2}} \quad (5)$$

where:

- φ_{robot} – Robot navigation direction
- ψ_{target} – Target direction
- λ – Intensity of repulsive force
- δ – Range of repulsion

The obstacle repulsive force magnitude depends on the distance to the robot. It can be modelled by the following equation:

$$\lambda = \beta_1 \cdot e^{-\frac{d}{\beta_2}} \quad (6)$$

where:

- β_1 – Maximum intensity of the repulsive force
- β_2 – Decline rate of the repulsion intensity with the distance to the obstacle
- d – Distance to the obstacle

The repulsion range depends on the size of the object, the bigger the obstacle the greater the range (Fig. 3).

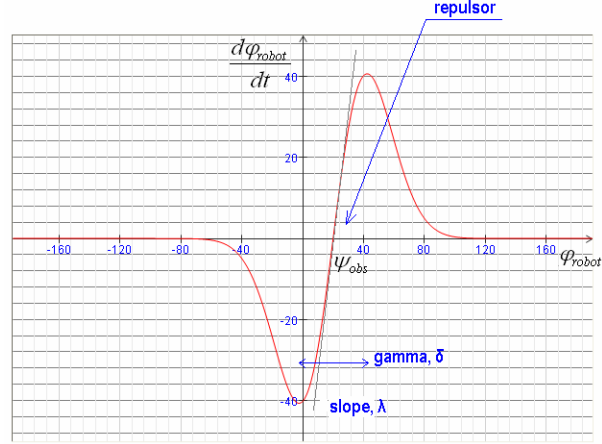


Fig. 3. Repulsive force as function of obstacle presence

Following the same technique as the target pursuing, as the robot moves also a repulsor fixed point shift occurs in the navigation direction. This brings up a repulsive force that compels the system to move away from this value due to the slope's negative value of the tangent to the curve at this point. As shown in Fig. 4, the intensity of the repulsive force varies according to the obstacle distance.

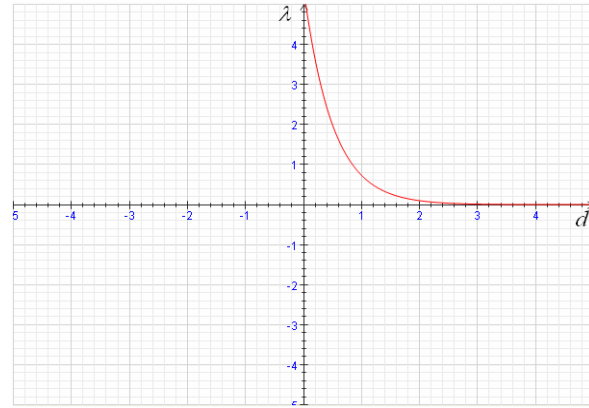


Fig. 4. Intensity of the repulsive force varying according to the obstacle distance

In this case, the behaviour variable is the angle difference between the robot direction and the target angle. The functions corresponding to obstacles will be continuously shifting along the same axis depending on their relative angles to the robot.

In the same way, the direction variation rate in time corresponds to the angular velocity or rotational movement of the robot.

The differential equation assumes the form:

$$\omega(t) = \frac{d\varphi_{robot}}{dt} = -\lambda \cdot (\Delta\alpha - \Delta\beta) \cdot e^{-\frac{(\Delta\alpha - \Delta\beta)^2}{2\delta^2}} \quad (7)$$

where:

ω – Robot's Angular velocity

$\Delta\alpha$ – Target angle relative to the robot

$\Delta\beta$ – Difference between obstacle and target angles

As it is possible to get an unlimited number of obstacles in an unknown environment, the global corresponding function is given by the sum of each of the individual forces, which results in the following equation:

$$\omega(t) = \frac{d\varphi_{robot}}{dt} = \sum_{i=1}^n -\lambda_i \cdot (\Delta\alpha - \Delta\beta_i) \cdot e^{-\frac{(\Delta\alpha - \Delta\beta_i)^2}{2\delta^2}} \quad (8)$$

where:

n – Number of obstacles

The repulsive intensity to the obstacle, i , is given by the equation:

$$\lambda_i = \beta_1 \cdot e^{-\frac{d_i}{\beta_2}} \quad (9)$$

The robot behaviour is based on the sum of all forces, including obstacles repulsive forces and the target attractive force. The resulting curve can assume many forms, as the one in Fig. 5. In this particular example, besides the target there are two obstacles placed in different directions. The resulting sum is a dynamic non-linear function (Fig. 6).

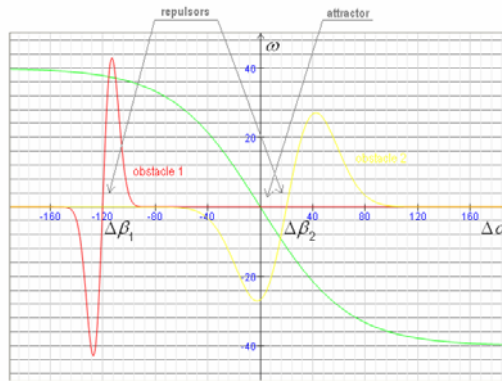


Fig. 5. Target and 2 obstacles vector fields' representation

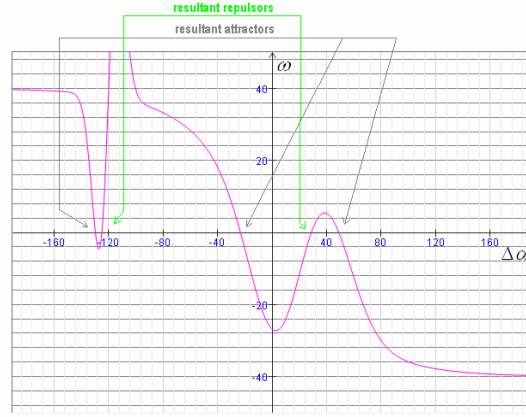


Fig. 6. Resultant vector field to the navigation direction

Depending on the robot navigation direction, the system may adopt any of the attractor fixed points, stepping away from the repulsors. As robot moves, all vector fields vary, and attractor and repulsive points appear and disappear in a dynamic way. This is an adaptive control to an unknown environment. The self decision is dynamic, so the behaviour is very smooth, similar to human decisions.

3. Linear Velocity Control

The robot linear velocity formula is continuously generating values so that the robot moves in a straight line to the target. This series of values, $x_{robot}(t)$, are generated by a dynamic system formulated by the following differential equation:

$$\frac{dx_{robot}}{dt} = f_{target}(x_{robot}) \quad (10)$$

The linear velocity control dynamic system differential equation, is as follows:

$$v(t) = \frac{dx_{robot}}{dt} = -v_{max} \cdot \tanh\left(\left((x_{robot} - x_{target}) - \sigma\right) \cdot \lambda\right) \quad (11)$$

where:

- v – Robot linear velocity
- x_{robot} – Robot position in a straight line to the target
- v_{max} – Robot maximum linear velocity
- x_{target} – Target position in a straight line to the robot
- σ – Security distance to the target
- λ – Attraction intensity to the target

Note: $\Delta d = x_{robot} - x_{target}$, (distance to target)

As it can be seen in **Fig. 7**, when the difference between the robot and target position is equal to the safety distance the robot must stop.

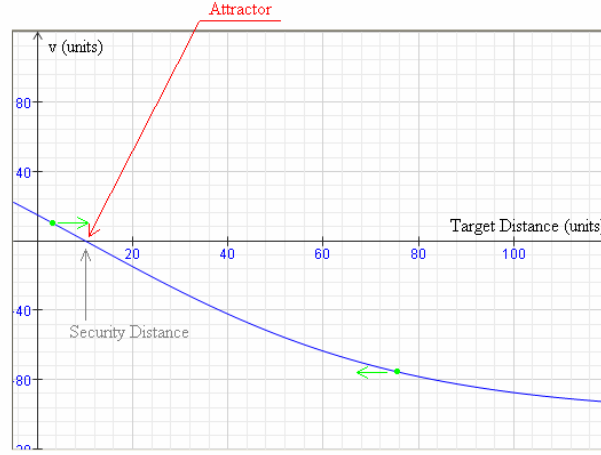


Fig. 7. Robot linear velocity graphical representation

The linear velocity indicated on the chart (v), is an input parameter to the function of robot motors and ranges from 0 to 100 (there is no specific unit). The distance to the target also ranges between 0 and 100.

As in the rotational movement, one important parameter is λ , because it allows specifying the target attraction intensity, also known as robot response time. But this value must not be very high otherwise the system becomes unstable.

The equation implemented on the robot and represented in figure 8 is:

$$\omega(t) = -100 \cdot \tanh((\Delta d - 10) \cdot 0.015) \quad (12)$$

4. Experimental Results

The robot's behaviour, using dynamic control, was first tested by creating a moving target (ball) in an environment without obstacles. It should be noticed that robot orientation, obstacles and target angles as well as respective distances were obtained through vision software. Fig. 8 shows results of rotational velocity variation in terms of target angle. The reference angle 180° means the robot is facing the target.

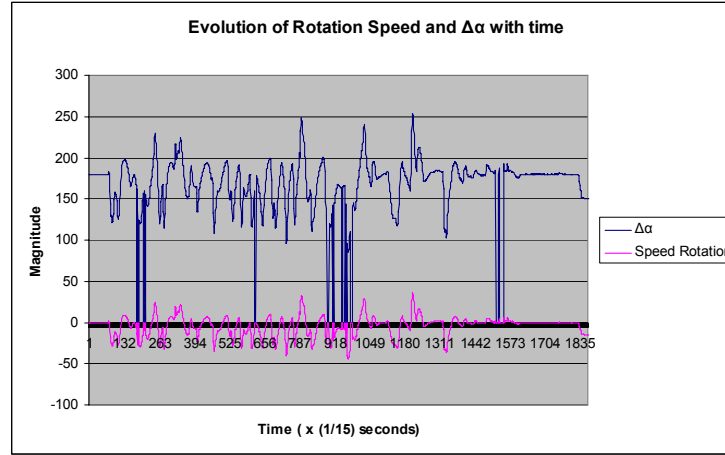


Fig. 8. Rotational velocity and corresponding angle between ball and robot

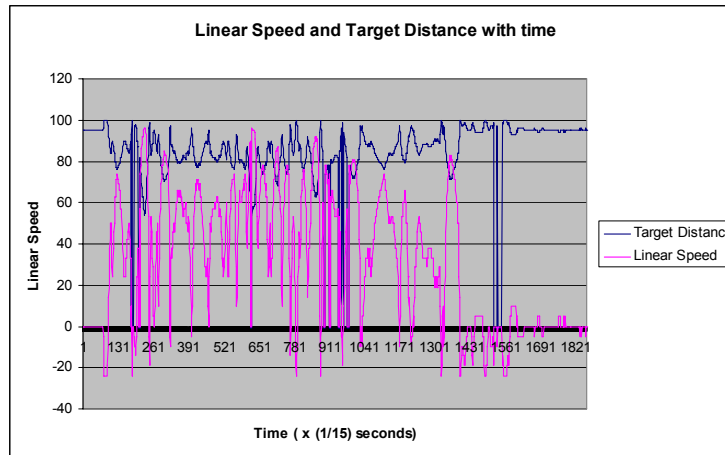


Fig. 9. Robot linear velocity versus target distance

The variation of linear velocity is represented in Fig. 9 as function of time and distance to the target. The value "95" is a reference position indicating that the robot reached the target. When the target moves away from the robot this value decreases and vice-versa. One unit of time corresponds to one frame. The processing cycle time is about 66ms.

5. Conclusions

From the experimental results shown above it can be concluded that the robot responds very fast to changes, and hence maintaining a smooth behaviour through time. This type of control doesn't need calibration parameters as opposite to PID control approach previously used [³]. Besides, the battery charge does not have any influence on the robot movement performance as happened with PID control algorithms.

This fast and stable dynamic control provides a very good precision and suits very well applications where low time-consuming algorithms are required for navigation in unknown environments.

Soon the complete control algorithm will be tested with moving obstacles included. Theoretically it is a very good adaptive control model.

The team would like to thank Prof. Estela Bicho for her support.

¹ "The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform", E. Bicho, G. Schöner, *Robotics and Autonomous Systems* 21 (1997) pp.23-35.

² "Target representation on an autonomous vehicle with low level sensors", E. Bicho, P. Mallet, G. Schöner, *The International Journal of Robotics Research*, Vol. 19, No. 5, May 2000, pp. 424-447.

³ Fernando Ribeiro, Ivo Moutinho, Pedro Silva, Carlos Fraga, Nino Pereira, "Vision, Kinematics and Game strategy in Multi-Robot Systems like MSL RoboCup", *ROBOCUP'2004 – Team Description Paper*, July 2004.